

# Nintendo GameCube™ Reset Guidelines

Version 1.41

## Table of Contents

|   |          |
|---|----------|
| <b>1. About Reset .....</b>   | <b>3</b> |
| <b>2. Reset Procedures.....</b>   | <b>3</b> |
| 2.1 Hot reset.....  | 3        |
| 2.2 Restart.....  | 3        |
| 2.3 Application-driven Restart .....  | 3        |
| <b>3. API .....</b>   | <b>3</b> |
| 3.1 OSResetSystem.....  | 3        |
| 3.2 OSGetResetCode.....   | 4        |
| 3.3 OSGetResetSwitchState.....  | 4        |
| 3.4 OSSetResetCallback.....   | 4        |
| <b>4. Processes Prior to Reset.....</b>   | <b>4</b> |
| 4.1 Reset the Nintendo GameCube™ Controller's origin value .....                            | 4        |
| 4.2 Stop the Nintendo GameCube™ Rumble Feature .....  | 4        |
| 4.3 Wait for end of writing to Nintendo GameCube™ Memory Card.....                          | 4        |
| 4.4 Wait for interruption/completion of graphics and audio .....                            | 4        |
| 4.4.1 Hot reset and restart .....   | 5        |
| 4.4.2 Application-driven Restart .....  | 5        |
| 4.5 Performing a Game Disc Check (only when performing an Application-driven Restart) ..... | 5        |
| <b>5. Cautions.....</b>   | <b>6</b> |
| 5.1 Handling the RESET Button.....  | 6        |
| 5.2 RESET Button chattering.....  | 6        |
| 5.3 Countering erroneous RESET Button press interrupts due to external noise.....           | 6        |
| 5.4 Resetting with operation other than RESET Button.....                                   | 6        |
| 5.5 Calibrate origin when RESET Button is pressed.....                                      | 6        |
| 5.6 Performing a Game Disc Check (only when performing an Application-driven Restart) ..... | 6        |
| <b>Appendix A. Reset Flow .....</b>   | <b>7</b> |
| A.1 Hot reset .....   | 7        |
| A.2 Restart .....   | 8        |
| A.3 Application-driven Restart.....   | 9        |

**Revision History**

| Version | Revision Date | Description of Revisions  |
|---------|---------------|---|
| 1.41    | 8/31/01       | - Revised paragraphs 1, 3.1, 4.1, 4.2, 4.4.2, 4.5, 5.1, 5.3, and 5.5<br>- Revised Appendix A  |
| 1.4     | 8/3/01        | - Changed "system reset" to "hot reset" throughout<br>- Added a third reset process "Restart" (revisions to paragraphs 1, 2, 3.1, 4.4.1, 4.5, Appendix A) |
| 1.0     | 7/13/01       | - Released by NOA   |

## 1. About Reset

This document provides explanations and warnings related to the process of resetting the Nintendo GameCube™. In this context, "reset" refers to the process that returns the game to the initial screen. The next section explains three ways of preparing an application process for resetting the game. The Nintendo GameCube™ has a RESET Button, but the application is responsible for the process that is performed when this button is used. Choose the method that is most appropriate for your game. The "reset" operation described in this document refers to both a reset that is done by pressing and releasing the Nintendo GameCube™ RESET Button, and a reset performed with the Nintendo GameCube™ Controller (in-hand Reset).

## 2. Reset Procedures

There are three reset procedures:

- Hot reset
- Restart
- Application-driven restart

This section explains all of these reset procedures. No matter which reset procedure you use, there are certain steps that must be performed before the reset process is executed. Please refer to Section 4.

### 2.1 Hot reset

If you specify `OS_RESET_HOTRESET` as the value for *reset* in the `OSResetSystem` function's first argument, the hardware will be reset while the application is running. By performing a hot reset, the system first returns to the IPL before reloading the executable file (the .dol file) from the Nintendo GameCube™ Game Disc and then restarts the application. Additionally, if you specify `TRUE` as the value for *forcemenu* in the `OSResetSystem` function's third argument, you can go directly to the IPL's menu screen.

For details, see the section on the `OSResetSystem` function in the Nintendo GameCube™ Function Reference Manual.

### 2.2 Restart

If you specify `OS_RESET_RESTART` as the value for *reset* in the `OSResetSystem` function's first argument, the system reloads the executable file (the .dol file) from the Nintendo GameCube™ Game Disc and then restarts the application.

For details, see the section on the `OSResetSystem` function in the Nintendo GameCube™ Function Reference Manual.

### 2.3 Application-driven Restart

This procedure returns the game process to a predetermined position. The exact position is up to the programmer: it can be the opening movie, or the game start screen or some other place. The programmer is also responsible for such processes as saving the variables that were used during the game.

## 3. API

Below is a summary of the functions that have been prepared for the reset procedure. For details, please read the Nintendo GameCube™ Function Reference Manual.

### 3.1 OSResetSystem

This function performs either a hot reset or a restart for the hardware. Select which of these you want to perform in *reset*, contained in the first argument of the function. If you choose to perform a hot reset, you can go directly to the IPL's menu screen by also specifying `TRUE` for *forcemenu* in the third argument of the function.

Caution: Since hot reset takes more than a few seconds through the IPL splash screen until the game program restarts, please avoid use of this function wherever possible. The hot reset is appropriate only in the following two situations.

**There is not enough space on the Memory Card**

Applications cannot erase files created by another application. To solve this problem, you can ask the game player if they want to go to the IPL's menu screen and erase unneeded files. In this situation you need to set ***forcemenu*** to TRUE and perform a hot reset. The IPL goes to menu mode automatically making it easy to use. (A function is used to move the IPL to menu mode).

**Operations for an application-driven restart are being used, a reset operation is performed, and the result of the "Disc Check" is not normal. (See Section 4.5)**

In this situation, do the opposite of the preceding and set ***forcemenu*** to FALSE.

### **3.2 OSGetResetCode**

This function returns the reset code for the reset performed by OSResetSystem.

### **3.3 OSGetResetSwitchState**

This function gets the state of the RESET Button, which determines whether the button is pressed or not.

### **3.4 OSSetResetCallback**

This registers the callback function for the reset process.

Caution: This function is used for debugging. Do not use the function OSSetResetCallback in your applications to detect pressing of the RESET Button. This is because a RESET Button press interrupt can be erroneously generated at the same time that the Controller is plugged into Nintendo GameCube™.

## **4. Processes Prior to Reset**

By implementing appropriate processes before the reset is executed, you can ensure that the game player is not inconvenienced by the reset. Below are several required and recommended processes to perform prior to executing the reset.

### **4.1 Calibration of the Nintendo GameCube™ Controller**

If the game player carries out a reset operation, the origin values for the Controller must also be recalibrated. Use the PADRecalibrate function to recalibrate the origin values. Additionally, the origin values for the Controller are recalibrated when the OSResetSystem function is called.

### **4.2 Stop the Nintendo GameCube™ Rumble Feature**

If a Controller placed on a tabletop is allowed to continue to vibrate after the reset process has been executed, it could end up falling off the table. To avoid this from happening, the Rumble motor must be halted before the reset process is completed. The PADRecalibrate function halts the rumble motor in addition to recalibrating the origin of the Controller. The Rumble motor is also halted when the OSResetSystem function is called.

### **4.3 Wait for end of writing to Nintendo GameCube™ Memory Card**

If the reset process is inadvertently executed while data is being written to the Memory Card, the integrity of the data will be lost. When data needs to be written to numerous files in one save process, wait until all the writing processes have been completed before executing the reset process.

### **4.4 Wait for interruption/completion of graphics and audio**

It is not good for the reset process to disturb the graphics and audio. It is recommended that you implement appropriate measures to fade out graphics and sound. In the sections below, explanations are given for some specific processes for each type of reset.

#### 4.4.1 Hot reset and restart

When you perform a reset using the `OSResetSystem` function, there is a possibility that the graphics functions will display unintended results. Therefore, perform resets from the `OSResetSystem` function only after the graphics functions have completed their operations.

However, audio functions will be terminated by the `OSResetSystem` function. As far as audio functions are concerned, we recommend terminating the audio functions from the application prior to executing `OSResetSystem`.

#### 4.4.2 Application-driven Restart

For the application-driven restart process, you can just return to the game start screen in a fashion similar to moving to a new stage in the game. However, be sure to recalibrate the origin values for the Controller.

#### 4.5 Performing a Game Disc Check (only when performing an Application-driven Restart)

It is necessary to perform a Game Disc check before performing an application-driven restart. Use the `DVDCheckDisk` function to check whether the Disc Cover was opened. If it is discovered that the Disc Cover was opened, it is necessary to perform a hot reset instead of an application-driven restart.

Here's what happens if you don't perform a Game Disc check. Consider a situation in which game A is being played and the player switches to a Game Disc for game B, then presses and releases the RESET Button. The game performs an application-driven restart. At this point the user expects game B to start. If game A uses an application-driven restart process and a Game Disc check is not performed, the game may continue to run game A, despite the fact that game B is now loaded. This will cause error messages, and result in the player being unable to play game B.

Therefore, always perform a Game Disc check before you perform an application-driven restart.

If a player opens and closes the Disc Cover without changing the Game Disc and tries to perform an application-driven restart, a Hot reset must be performed instead. This is required because it takes as much as 8 seconds to perform the Game Disc check, and an additional 8 seconds to perform a Hot reset. Therefore, if a new Game Disc is detected, the player will have to wait up to 16 seconds before game play resumes.

Additionally, if the Disc Cover is opened during the game, the device driver will automatically determine if the correct Game Disc is inserted when the next access is performed (this takes approximately 8 seconds). If the timing of the RESET Button being pressed and released is after confirmation of a correct Game Disc, the Game Disc check returns a "Recognized" result so there is no need for a hot reset. In this manner, even after the "final" determination by the device driver is done, a verification of whether or not the Disc Cover has been opened even once can be done.

Use the `DVDCheckDisk` function for Game Disc checks.

NOTE: In the case of a "Restart" (first argument `OS_RESET_RESTART`) or Hot Reset (first argument `OS_RESET_HOTRESET`), the `OSResetSystem` function performs a Game Disc check and switches to a Hot Reset if the Game Disc check doesn't return "recognition completed". Therefore, you do not need to perform a Game Disc check prior to executing `OSResetSystem`.

The intended purpose here is not to recommend a hot reset be carried out every time the Disc Cover is opened. The intent is for you to perform a hot reset when the RESET Button has been pressed and released, **and** the Game Disc has not been recognized.

## **5. Cautions**

### **5.1 Handling the RESET Button**

Game players may push the RESET Button when they want to restart a game or when they feel that the software is not operating properly. Please perform reset process when the RESET Button is pressed and released. Do not perform any operation other than a reset when the RESET Button is used.

### **5.2 RESET Button chattering**

In this context, chattering refers to the detection of multiple input signals when the RESET Button is pressed only once. RESET Button chattering is handled internally by the OSGetResetSwitchStat function. If you apply the reset process at the exact time that the RESET Button is pressed, then chattering could result in the application of multiple reset processes. Thus, please do not apply the reset process until after checking that the player has released his or her finger from the RESET Button.

### **5.3 Countering erroneous RESET Button press interrupts due to external noise**

Unanticipated external noise can lead to the erroneous generation of a RESET Button press interrupt by the Nintendo GameCube™. For example, this problem has been confirmed to occur when the Controller is plugged into the Nintendo GameCube™.

The function OSSetResetCallback is linked to a RESET Button press interrupt, and it will call the registered callback. But if the reset process is started from the registered callback, there is a danger that the reset process will start even though the player has not touched the RESET Button. For this reason, please do not use the function OSSetResetCallback to detect a pressing of the RESET Button. (However, using the function OSSetResetCallback for debugging with the Nintendo GameCube™ Software Development Kit poses no problems.)

To access the state of the RESET Button within the application, please perform polling using the function OSGetResetSwitchState. This function performs noise countermeasures internally, so the application does not need to implement noise countermeasures of its own. Please use the function OSGetResetSwitchState in every game frame so there will be no delay in a reset operation initiated by the game player.

### **5.4 Resetting with operation other than RESET Button**

The Nintendo GameCube™ has a RESET Button that is meant for operating the reset process. However, for game player convenience you can also add a separate reset operation that is performed with the Nintendo GameCube™ Controller (an in-hand reset). This additional reset operation can be executed by selecting a menu item from inside the game, or by establishing a reset-use Controller command (B Button, X Button and START/PAUSE Button pressed simultaneously for longer than 0.5 seconds).

### **5.5 Calibrate origin when reset operation is performed**

Game players will sometimes perform a reset operation in order to calibrate the Controller. Be certain to execute a recalibration of the Controller's origin values whenever the game player performs a reset operation. However, note that if the OSResetSystem function is being used this recalibration is done internally, so no specific precautions are required.

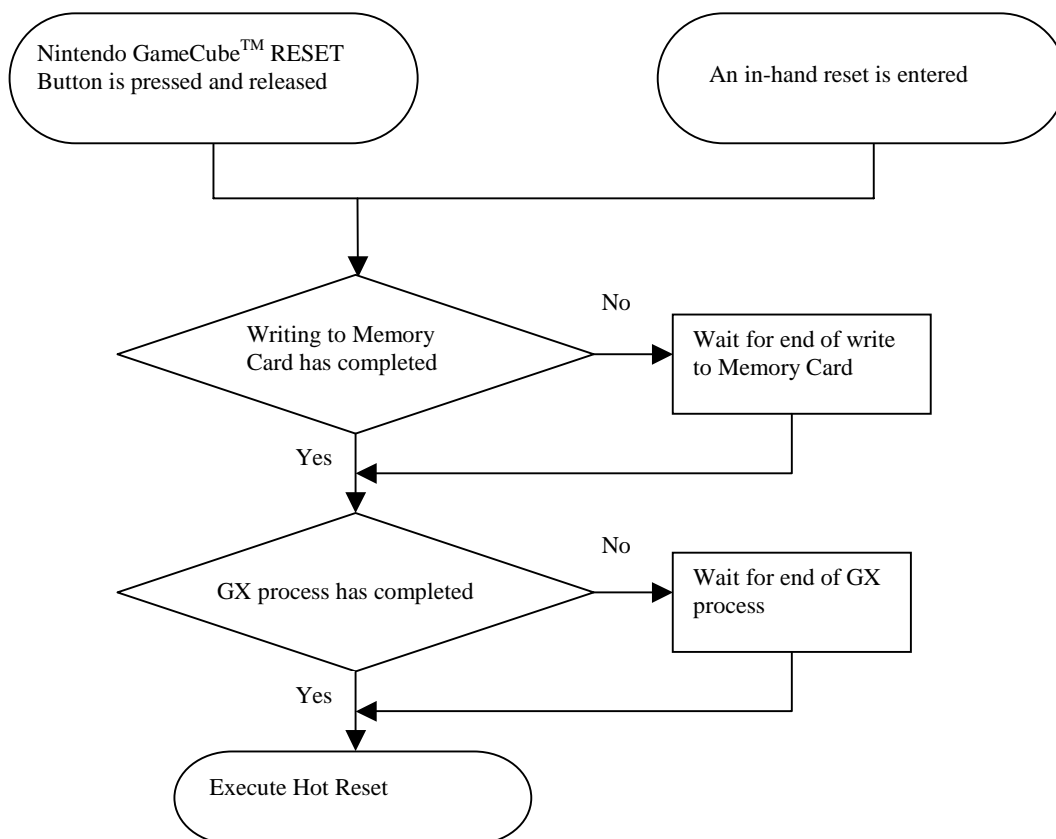
### **5.6 Performing a Game Disc Check (only when performing an Application-driven Restart)**

If you perform an application-driven restart, always perform a Game Disc check before beginning the restart process. For details, see paragraph 4.5 "Performing a Disc Check (Only when Performing an Application-driven Restart)".

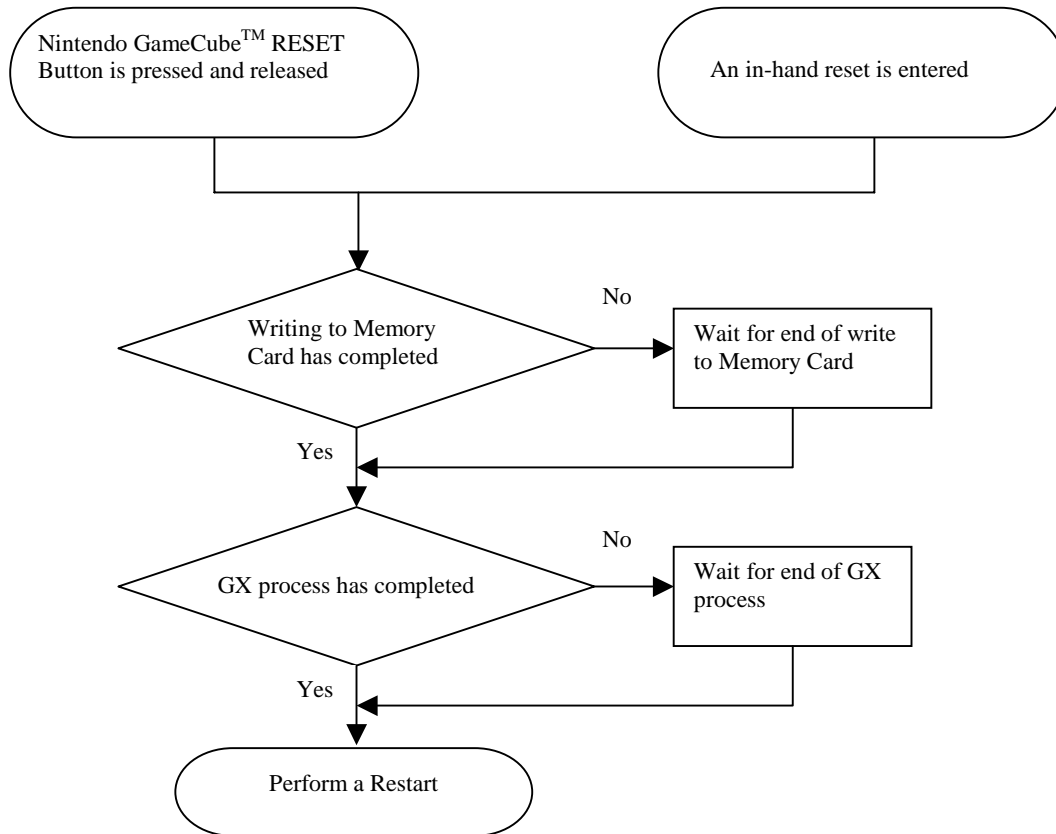
## Appendix A. Reset Flow

This section uses schematic diagrams to explain the process flow for a Hot reset, Restart, and for an application-driven restart. Note that for reset operations that involve the Nintendo GameCube™ RESET Button, the reset process is executed after confirming that the RESET Button has been depressed and then released (see Section 5.2).

### A.1 Hot reset



## A.2 Restart





### A.3 Application-driven Restart

The flow diagrammed here is just one example of how the process can run. For details, please read the various cautions described in the previous section

